

Learning quantum Hamiltonians from single-qubit measurements

Liangyu Che,^{1,2,*} Chao Wei,^{1,2,*} Yulei Huang,^{1,2} Dafa Zhao³,, Shunzhong Xue,³ Xinfang Nie,^{1,2} Jun Li,^{1,2,†} Dawei Lu³,, and Tao Xin^{1,2,§}

¹Shenzhen Institute for Quantum Science and Engineering and Department of Physics, Southern University of Science and Technology, Shenzhen 518055, China

²Guangdong Provincial Key Laboratory of Quantum Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China

³State Key Laboratory of Low-Dimensional Quantum Physics and Department of Physics, Tsinghua University, Beijing 100084, China



(Received 7 January 2021; revised 5 April 2021; accepted 29 April 2021; published 25 June 2021)

In the Hamiltonian-based quantum dynamics, to estimate Hamiltonians from the measured data is a vital topic. In this work, we propose a recurrent neural network to learn the target Hamiltonians from the temporal records of single-qubit measurements, which does not require the ground states and only measures single-qubit observables. It is applicable on both time-independent and time-dependent Hamiltonians and can simultaneously capture the magnitude and sign of Hamiltonian parameters. Taking the Hamiltonians with the nearest-neighbor interactions as numerical examples, we trained our recurrent neural networks to learn different types of Hamiltonians with high accuracy. The study also shows that our method has good robustness against the measurement noise and decoherence effect. Therefore, it has widespread applications in estimating the parameters of quantum devices and characterizing the Hamiltonian-based quantum dynamics.

DOI: [10.1103/PhysRevResearch.3.023246](https://doi.org/10.1103/PhysRevResearch.3.023246)

I. INTRODUCTION

Developing the methods for estimating Hamiltonians has two important motivations. First, Hamiltonians fully govern the dynamics of quantum systems, which determines whether the control operations are highly accurate on these quantum devices. For instance, quantum circuits are generally realized through the control pulse techniques [1], which are beforehand designed and optimized according to the estimated Hamiltonians. Second, as a branch of quantum process tomography [2], estimating Hamiltonians provides an alternative approach to characterize the performed quantum simulations. Third, estimating Hamiltonians can be also used to find the simplified models, which capture the physics of more complex quantum systems. It has potential applications in accelerating quantum simulations and gaining physical insight into the low-energy physics of a system. So far, some methods have been developed to achieve this purpose, including the self-learning Monte Carlo method to speed up simulations [3–7] and renormalization group-inspired machine-learning approach based on the partition function to optimize the effective models [8]. Therefore, estimating Hamiltonians is a

central problem in the quantum platforms [9], quantum control [10,11], and quantum simulations [12].

In principle, Hamiltonians can be estimated by quantum state and process tomography by considering the Hamiltonians are the generators of the dynamical processes [2,13,14]. However, this approach requires exponential physical resources, although many-body Hamiltonians have the polynomial number of unknown parameters because of the physical constraints. So far, various methodologies have been studied for this purpose. For instance, some methods using Fourier transform or fitting on the temporal records of measurement of some observables also are proposed to estimate Hamiltonians with few qubits [15–17]. Zhang and Sarvoar [18,19] proposed an approach for estimating Hamiltonians based on the limited measurements by the eigensystem realization algorithm (ERA). This method was experimentally demonstrated on nuclear magnetic resonance quantum processor [20]. Akira Sone *et al.* further studied the identifiability problem of Hamiltonians and the necessary experimental resources in ERA method [21,22]. Many-body local Hamiltonians can be uniquely estimated by a single eigenstate of Hamiltonians, which also inspires the subsequent research [23–27]. Recently, a quantum quench method was proposed to reconstruct a generic many-body local Hamiltonian [28], which uses pairs of generic initial and final states connected by the time evolution of Hamiltonians.

Machine learning has obtained great successes in solving the problems in quantum physics [29–40] such as the identification of quantum phase transitions [29–31], the classification of quantum topological phases and quantum entanglement [32–35], quantum state measurement and tomography [36–38]. Recently, machine learning also started

*These authors contributed equally to this work.

†lij3@sustech.edu.cn

‡ludw@sustech.edu.cn

§xint@sustech.edu.cn

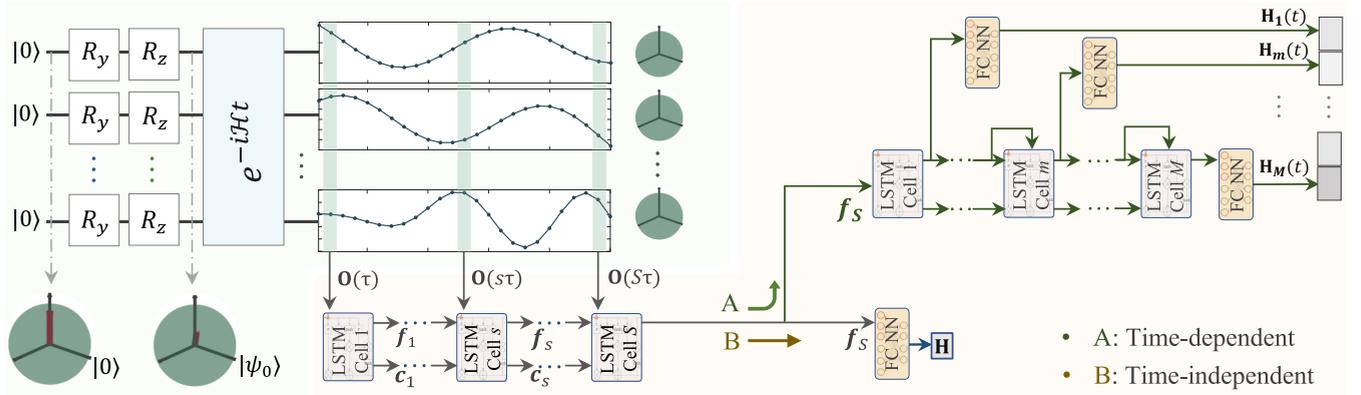


FIG. 1. Circuit diagram of our neural networks on learning the parameters of Hamiltonians from the temporal records of single-qubit measurements. We first perform the dynamical evolution $e^{-i\mathcal{H}t}$ from the initial state $|\psi_0\rangle$ (bottom left Bloch sphere). At each moment $s\tau$, we measure the expectation values of single-qubit Pauli operators (middle Bloch sphere), and they are collected as a vector $\mathbf{O}(s\tau)$ fed into the s th LSTM cell. Lastly, the combination of FC and LSTM neural networks for time-dependent parameters in Hamiltonians (path A) or an FC neural network for time-independent parameters in Hamiltonians (path B) follows LSTM cells.

to bring developments in estimating Hamiltonians. Reference [27] presents the deep neural network to recover 2-local Hamiltonians from merely 2-local measurements of ground states. Reference [41] proposes that convolutional neural networks can also be used to predict the physical parameters of Hamiltonians from the ground states. However, these methods usually require the assumption of the ground states.

In this work, we propose a machine-learning method, recurrent neural network (RNN), to learn Hamiltonians from single-qubit Pauli measurements on each qubit. The initial state does not require the ground states of target Hamiltonians and only single-qubit Pauli observables are measured at a discrete-time forming the temporal records of single-qubit measurements, which are fed into RNN. The intuition of this method is that if the Hamiltonians are identifiable under the temporal records of single-qubit measurements, then there exists the underlying rule from single-qubit measurements to the target Hamiltonians, which can be learned directly from single-qubit measurements via data-driven machine learning, although this rule may have complicated or even unknown functional forms.

II. MACHINE-LEARNING FRAMEWORK

Our machine-learning method is illustrated in Fig. 1. A N -qubit system starts from the initial state $|\Psi_0\rangle = \prod_{i=1}^N |\psi_0^i\rangle$ and undergoes a dynamical process governed by the unknown Hamiltonian,

$$\mathcal{H} = \sum_{m=1}^M a_m B_m, \quad (1)$$

where B_m is the tensor product of Pauli matrices I , σ_x , σ_y , and σ_z , and a_m is the parameter of Hamiltonians. For single-qubit Pauli operator $P \in S_P = \{\sigma_k^{(i)} | k = x, y, z, 1 \leq i \leq N\}$, its expectation value is $\bar{P}(t) = \langle \Psi_0 | P(t) | \Psi_0 \rangle$ with $P(t) = e^{i\mathcal{H}t} P e^{-i\mathcal{H}t}$. Here, $P(t) = \sum_{n=0}^{\infty} \frac{i^n t^n}{n!} P_n$. $P_0 = P$ and $P_n = \sum_{m=1}^M a_m [B_m, P_{n-1}]$. It shows that the temporal records of single-qubit measurements are polynomial equations of unknown parameters a_m 's in Hamiltonians, and the coefficients

of the equations are the expectation values of commute operators between single-qubit observables P and Hamiltonian terms B_i at the initial state $|\Psi_0\rangle$ (see Appendix A for details). If these coefficients are not zeros, the parameters will participate in the dynamics of single-qubit observables and it is possible to learn the Hamiltonian from the temporal records of the expectation values of P . In this situation, we say that the dynamics of single-qubit observables have nontrivial initial values. Here, we set $|\psi_0^i\rangle = R_z(\pi/4)R_y(\pi/4)|0\rangle$, which can be prepared from the state $|0\rangle$ using rotation operations $R_z(\pi/4)$ and $R_y(\pi/4)$. It is worth emphasizing that not all the Hamiltonians can be estimated via this approach, but most common of Hamiltonians are identifiable under single-qubit measurements and the initial state $|\Psi_0\rangle$. More discussions can be found in Sec. V.

During the dynamical evolution $e^{-i\mathcal{H}t}$, the expectation values of single-qubit operators $\sigma_{x,y,z}^{(i)}$ are measured at a discrete-time by time interval τ . Total sample points is denoted by S and then total sample time is $T = S\tau$. The temporal records of single-qubit measurements are collected as a vector,

$$\mathbf{I} = \{O_k^{(i)}(s\tau) | O_k^{(i)}(s\tau) = \text{Tr}(\rho(s\tau) \cdot \sigma_k^{(i)}), \quad (2)$$

$$1 \leq s \leq S, k = x, y, z, 1 \leq i \leq N\}.$$

$\rho(s\tau)$ is the density matrix of the system at the moment $s\tau$. The parameters of Hamiltonians are collected as a vector $\mathbf{H} = \{a_m | 1 \leq m \leq M\}$. Then we train a neural network framework consisting of long short-term memory (LSTM) and fully connected (FC) neural networks with generated training data $\{\mathbf{I}, \mathbf{H}\}$. In this training, the measured data $\mathbf{O}(s\tau) = \{O_k^{(i)}(s\tau)\}$ at moment $s\tau$ is fed into the s th LSTM cell. It is worth emphasizing that the structures for exporting \mathbf{H} are different for time-dependent and time-independent Hamiltonians. For time-dependent parameters in Hamiltonians, f_s is imported to a composite neural network including LSTM and FC neural networks (path A). For time-independent parameters in Hamiltonians, an FC neural network directly follows the LSTM cells. Here, if we only use one LSTM for learning the input and an FC neural network for generating the output

for time-dependent Hamiltonians, the time correlation in the output would be lost and the prediction accuracy will be low.

The objective function for one training sample is the mean square error (MSE) between \mathbf{H}^{pred} and \mathbf{H}^{true} as follows:

$$L_0 = \frac{1}{M} \sum_{m=1}^M (\mathbf{H}_m^{\text{true}} - \mathbf{H}_m^{\text{pred}})^2. \quad (3)$$

In our training networks, the input is the dynamics of single-qubit measurements \mathbf{I} , the output is the target Hamiltonian \mathbf{H} , and the optimization variables are the parameters of neural networks (weights W and bias b). We generate a lot of training samples and use Adam optimization algorithm to minimize $L(W, b) = \sum_{i=1}^{M_i} L_0^i(W, b)$ with the number of training samples M_i . Adam is one of the state-of-the-art stochastic gradient descent algorithm (SGD), which provides the analytical gradient in the minimization [42]. This algorithm uses momentum to suppress the oscillation, and by analyzing the gradient of the parameters, adaptively adjusts the learning rate to accelerate the convergence. Therefore, the convergence of this algorithm is faster than that of the traditional SGD algorithm in dealing with nonconvex optimization problems, and the solutions of Adam are also better than traditional SGD. After the training, we can predict the unknown Hamiltonian parameters \mathbf{H} from single-qubit measurements \mathbf{I} . Here, the definition of the cosine proximity function is adopted to estimate the accuracy of the prediction \mathbf{H}^{pred} by

$$F(\mathbf{H}^{\text{pred}}, \mathbf{H}) = \frac{(\mathbf{H}^{\text{pred}} \cdot \mathbf{H})}{(\|\mathbf{H}^{\text{pred}}\| \cdot \|\mathbf{H}\|)}. \quad (4)$$

More details about our NN can be found in Appendices B and C. Next, to demonstrate the feasibility of our machine-learning method, we consider a special class of Hamiltonians with the prior physical knowledge of the system and train neural networks to learn them.

III. EXAMPLES FOR LEARNING HAMILTONIANS

As the demonstrations of applications, we first train RNN framework for estimating time-independent Hamiltonians. First one is the Ising Hamiltonian \mathcal{H}_{XY}^N with the nearest-neighbor XY interactions placed in a static magnetic field around z axis as follows:

$$\mathcal{H}_{XY}^N = \sum_{i=1}^N a_z^{(i)} \sigma_z^{(i)} + \sum_{j=1}^{N-1} J^{(j)} (\sigma_x^{(j)} \sigma_x^{(j+1)} + \sigma_y^{(j)} \sigma_y^{(j+1)}). \quad (5)$$

Second one is the more general Ising Hamiltonian \mathcal{H}_{XYZ}^N with the anisotropic interactions in three directions,

$$\mathcal{H}_{XYZ}^N = \sum_{i=1}^N a_z^{(i)} \sigma_z^{(i)} + \sum_{i=1}^{N-1} (J_x^{(i)} \sigma_x^{(i)} \sigma_x^{(i+1)} + J_y^{(i)} \sigma_y^{(i)} \sigma_y^{(i+1)} + J_z^{(i)} \sigma_z^{(i)} \sigma_z^{(i+1)}). \quad (6)$$

$a_z^{(i)}$ and $J_{x,y,z}^{(j)}$ are the parameter of magnetic field on j th qubit and the coupling values between the nearest-neighbor qubits, respectively. Suppose $a_z^{(i)} \in [-J_0, J_0]$ and $J_{x,y,z}^{(j)} \in [-J_0, J_0]$. J_0 is a global factor, which is set to 1 in our training. In the training, the expectation values of single-qubit observables $\sigma_{x,y,z}^{(i)}$

are measured at a discrete-time separated by $\tau = 0.02\pi/J_0$ as the input data. The reason for choosing such a time interval can be found in the following section. For Eq. (5), we choose $N = 7$, $S = 25$, and generate 100 000 training data $\{\mathbf{I}, \mathbf{H}\}$ for training the neural networks. For Eq. (6), we choose $N = 6$ and $S = 75$; and 200 000 training data $\{\mathbf{I}, \mathbf{H}\}$ are generated for training the neural networks. After finishing the training of RNN on the training data, our RNN has the ability to estimate the unknown Hamiltonians from single-qubit measurements with high accuracy. Five thousand test data are generated to compute the similarity F_{test} between the actual parameters \mathbf{H}^{true} and the predicted outcome \mathbf{H}^{pred} . The averaged similarities on the whole test data are over 0.99 for \mathcal{H}_{XY}^7 and 0.98 for \mathcal{H}_{XYZ}^6 . Figure 2(a) gives our results, including L_{test} , L_{test} , and F_{test} as a function of epochs, and the comparison between the actual value $J_{\text{test}}^{(1)}$ and the prediction $J_{\text{pred}}^{(1)}$ for 100 randomly test data at the beginning and end.

Most of the existing methods are designed for the time-independent Hamiltonians and they are not directly applicable to time-dependent Hamiltonians. Our proposed RNN method presented in the above can also be used to learn time-dependent Hamiltonians. As a numerical demonstration, we consider a 3-qubit Hamiltonian $\mathcal{H}_{XY}^3(t)$,

$$\mathcal{H}_{XY}^3(t) = \sum_{i=1}^3 a_z^{(i)}(t) \sigma_z^{(i)} + \sum_{j=1}^2 J^{(j)} (\sigma_x^{(j)} \sigma_x^{(j+1)} + \sigma_y^{(j)} \sigma_y^{(j+1)}).$$

Here, we take $a_z^{(i)}(t)$ being a random combination of W Fourier series as an example. Actually, our method also works for other types of time-dependent Hamiltonians. $a_z^{(i)}(t) = \frac{1}{W} \sum_{w=1}^W F_w \cos(v_w t + \phi_w)$ and $J_i \in [-J_0, J_0]$ is static in time. $F_w \in [-J_0, J_0]$, $v_w \in [-J_0, J_0]$, and $\phi_w \in [0, 2\pi]$ are the amplitude, frequency, and phase of w th series, respectively. In this case, we set $W = 10$. The used neural network is presented in Fig. 1. The expectation values of single-qubit observables also are measured at a discrete time separated by $\tau = 0.02\pi/J_0$, and they are collected as a vector $\mathbf{I} = \{O_k^{(i)}(s\tau) | O_k^{(i)}(s\tau) = \text{Tr}(\rho(s\tau) \cdot \sigma_k^{(i)}), 1 \leq s \leq 300, k = x, y, z, \text{ and } 1 \leq i \leq 3\}$. The parameters of $\mathcal{H}_{XY}^3(t)$ are collected as a vector $\mathbf{H} = \{a_z^{(i)}(s\tau), J^{(j)} | 1 \leq s \leq 300, 1 \leq i \leq 3, 1 \leq j \leq 2\}$. Our training data also consists of 100 000 training data $\{\mathbf{I}, \mathbf{H}\}$. After training RNN to convergence on these training data, it can be used to learn the temporal behavior of $a_z^{(i)}(t)$ from only the measurements \mathbf{I} . Figure 3 presents the temporal behavior of the predicted values (solid lines) and its comparison with the actual values (dotted lines) for time-dependent parameters $a_z^{(i)}(t)$. It shows that a good agreement between the predicted and real results has been achieved.

IV. ROBUSTNESS OF PROTOCOL

It is known that learning from local observables may not yield good Hamiltonian models. For instance, the temporal records of single-qubit measurements inevitably are influenced by the statistical and environmental noises, and may these noises deviate the predicted values of RNN from the ideal ones. This may have undesired implications for the quantum control applications. For this purpose, we further numerically study the robustness of our RNN framework in

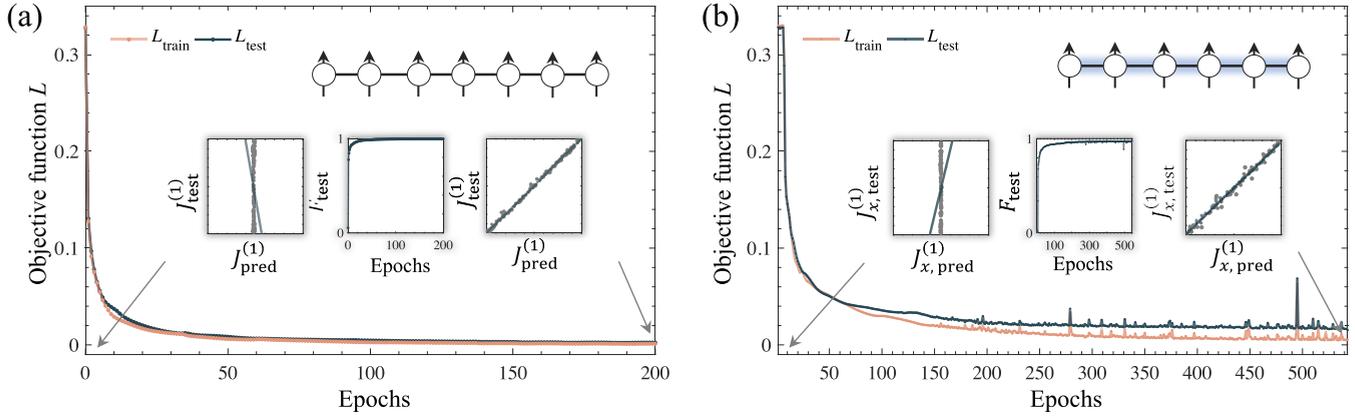


FIG. 2. Trained results for 7-qubit Ising Hamiltonian \mathcal{H}_{XY}^7 (a) and 6-qubit Ising Hamiltonian \mathcal{H}_{XYZ}^6 (b). The top right corners of panels respectively present their qubit configurations. The orange and cyan lines show the objective functions L_{train} and L_{test} as a function of epochs. The similarity F_{test} between the predicted \mathbf{H}^{pred} and the true \mathbf{H}^{true} in the test data is also presented with the increase of epochs (middle subfigures). At the beginning and end of the training, we randomly choose 100 samples and plot the comparison between the predicted and actual values for the parameters $J^{(1)}$ (left and right subfigures).

learning Hamiltonians under the Gaussian noise and decoherence effect by considering the Hamiltonian \mathcal{H}_{XY}^3 in Eq. (5).

First, we train RNN frameworks by feeding 100 000 noiseless training data $\{\mathbf{I}, \mathbf{H}\}$ with the sampling points $S = 25$ and $S = 50$, respectively, and then use them (RNN_0noise_25 and RNN_0noise_50) to learning Hamiltonians from the noisy test data. Figure 4(a) presents the predicted accuracy as a function of the noise level ϵ . RNN_0noise_50 has a better performance than RNN_0noise_25, but both their predicted accuracy decrease with the increasing of ϵ . ϵ is the standard deviation of the Gaussian noise. When $\epsilon = 0.1$, the accuracy of RNN_0noise_25 decreases to 0.98. To further improve the robustness of our RNN frameworks under the noise, we change to train RNN frameworks by feeding noisy training data under a Gaussian noise with $\epsilon = 0.1$. Similarly, we use these models to test the noisy data. The average values of

prediction accuracy as a function of ϵ also are presented in Fig. 4(a), which shows a good performance with the similarity of over 0.99 and the predicted accuracy improves to 0.995 from the previous 0.98 when $\epsilon = 0.1$. The above simulations show that training RNN frameworks with the noisy data will greatly enhance the predicted accuracy and the more sample points will bring better robustness against the noise.

Second, the total time for measuring the temporal records may reach or even exceed the coherence time of the experimental devices. Hence, the collected temporal records contain the decoherence effect, leading to a decrease in the predicted accuracy. For this purpose, we also numerically study the performance of our RNN frameworks under the decoherence effect. Similarly, we respectively use the noiseless and decoherence data ($\tau = 0.02\pi/J_0$, $S = 10$, and $S\tau = 3\pi/J_0$) to train two models (RNN_0noise_150 and RNN_T2noise_150) and then use them to learn Hamiltonians from the decoherence test data. Figure 4(b) shows that the prediction accuracy will have a significant improvement with the average value of over 0.99, when using RNN_T2noise_150 to process the

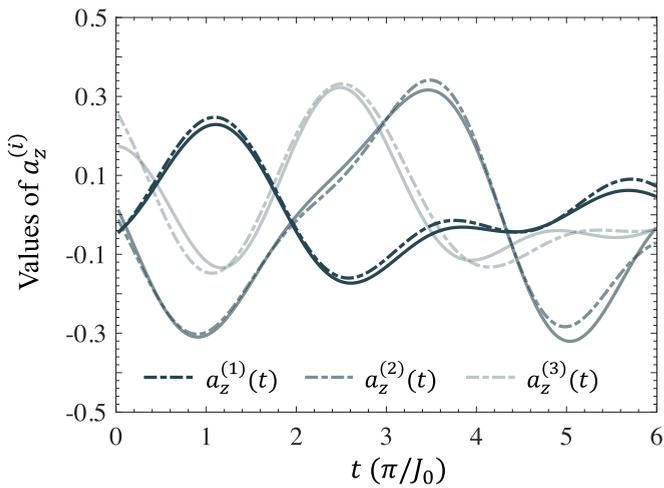


FIG. 3. The temporal curves of the actual parameters (dotted lines) and the values learned by RNN (solid lines) for time-dependent parameters $a_z^{(i)}(t)$. The predictions of time-independent parameters are $J_{\text{pred}}^{(1)} = 0.0464$ ($J_{\text{true}}^{(1)} = 0.0326$) and $J_{\text{pred}}^{(2)} = -0.0345$ ($J_{\text{true}}^{(2)} = -0.0181$).

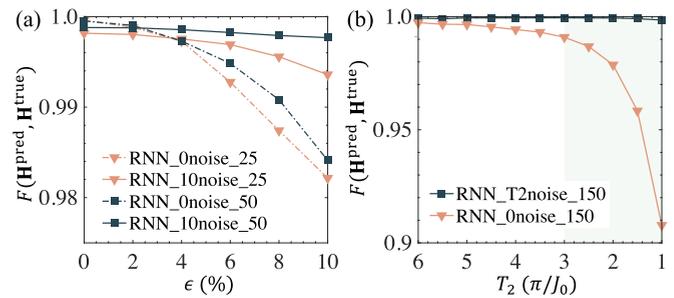


FIG. 4. The numerically simulated results for the robustness. (a) The predicted accuracy of the trained RNN models (RNN_0noise_25, RNN_10noise_50, RNN_0noise_25, and RNN_10noise_50) under the influence of Gaussian noise. (b) The predicted accuracy of trained RNN models (RNN_T2noise_150 and RNN_0noise_150) under the influence of decoherence effect. The cyan shadow is that the sampling time is longer than coherence time.

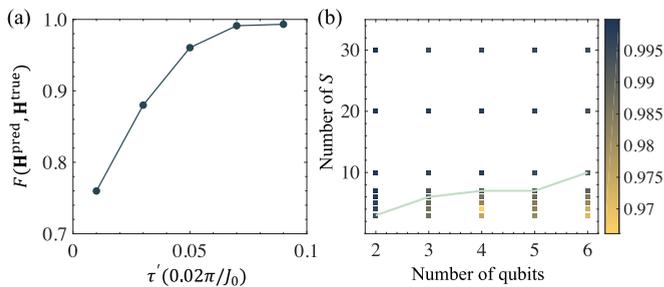


FIG. 5. Numerical simulations in the discussion. (a) The achieved accuracy with the different sampling intervals τ' and fixed sampling points $S = 25$. (b) The achieved accuracy under the different number of qubits N and sampling points S . The simulations are performed for the Hamiltonian \mathcal{H}_{XY}^3 . The cyan line is drew for the points with the accuracy of over 0.99.

decoherence test data. Therefore, the learned models via our method are at least accurate under the measurement error caused by the statistical and environmental noises. More details about simulations can be found in Appendix D.

V. DISCUSSIONS

We first analyze the sampling setting in our machine-learning method. (1) The sampling interval τ should be traded-off, accounting for the coherence time. On the one hand, the total sampling time may exceed the coherence time of qubits if τ is too large, leading to the decrease of the prediction accuracy. On the other hand, the temporal records of single-qubit measurements may be hard to distinguish if τ is too small, also leading to the decrease of the prediction accuracy. As shown in Fig. 5(a), we train the neural networks with different sampling interval τ' and fixed $S = 25$. The result shows that the RNN model cannot be trained to a high accuracy if τ' is too small. (2) The number of total sampling points is $3NS$, where factor 3 is the number of elements $\sigma_{x,y,z}^{(i)}$, N is the number of qubits, and S is the number of sample points. In Fig. 5(b), we train the neural networks under different N and S for the Hamiltonian \mathcal{H}_{XY}^3 . It is shown that S has a gentle increase with the size of the system for this type of Hamiltonians. It may be understood from the following aspect. As long as this Hamiltonian is identifiable under the chosen initial states and single-qubit observables, it is possible to learn the Hamiltonians from their temporal records with finite sampling points. For instance, many-body Hamiltonians have polynomial parameters. The polynomial sampling points may be enough to estimate the parameters for many-body Hamiltonians via machine-learning method.

In our machine-learning method, the nontrivial initial states are not unique. We choose the product state $|\Psi_0\rangle = R_x(\pi/4)R_y(\pi/4)|0\rangle^{\otimes n}$ as the initial state considering that the product states are easy to prepare in the experiments. Some entangled states can be also used as the initial states, for instance, we adopt the entangled state $|\Psi_0\rangle = R_x(\pi/4)R_y(\pi/4)\frac{|000\rangle+|111\rangle}{\sqrt{2}}$ as the initial state and a 3-qubit Hamiltonian \mathcal{H}_{XY}^3 as the model to test our method. The trained results are shown in the Fig. 6. It shows that the predicted values also match with the test values very well and $F_{\text{test}} = 0.998$.

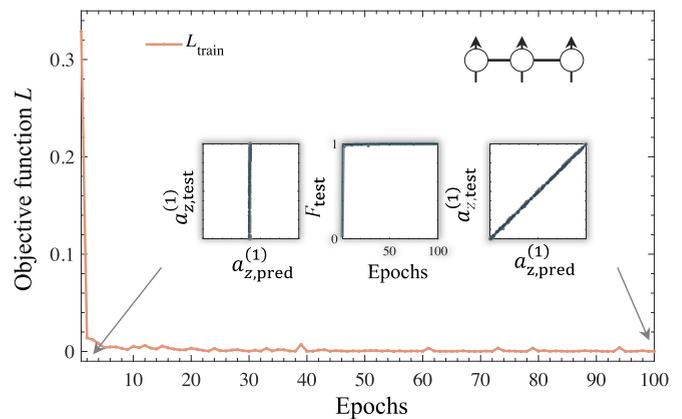


FIG. 6. Trained results for 3-qubit Hamiltonians \mathcal{H}_{XY}^3 starting from the initial state $|\Psi_0\rangle = R_x(\pi/4)R_y(\pi/4)\frac{|000\rangle+|111\rangle}{\sqrt{2}}$.

For many Hamiltonians that could reproduce the same local dynamics, our method does not learn the right Hamiltonians from only single-qubit measurements. Fortunately, most of the interested Hamiltonian models in quantum simulations, including but not limit to the examples studied in this paper, usually have one-to-one mapping because of the physical constraints. To extend the applications of our method, we attempt to learn the systems without any local magnetic orders, the Kitaev model on the honeycomb lattice, which is the interesting model in quantum spin liquid [43,44]. The corresponding Hamiltonian can be written as

$$\mathcal{H}_{\text{Kit}}^6 = J^{(1)}(\sigma_z^{(1)}\sigma_z^{(2)} + \sigma_z^{(4)}\sigma_z^{(5)}) + J^{(2)}(\sigma_y^{(2)}\sigma_y^{(3)} + \sigma_y^{(5)}\sigma_y^{(6)}) + J^{(3)}(\sigma_x^{(3)}\sigma_x^{(4)} + \sigma_x^{(1)}\sigma_x^{(6)}). \quad (7)$$

As shown in Fig. 7, the similarity between the predicted values and the test values remains high $F_{\text{test}} = 0.999$, which means our method has potential applications on the Hamiltonian model without any local magnetic orders. For this Hamiltonian model without any local magnetic terms, our method also works well. It can be understood in the following. Although we only measure the single-qubit measurements, the Hamiltonian parameters will also participate in the dynamics

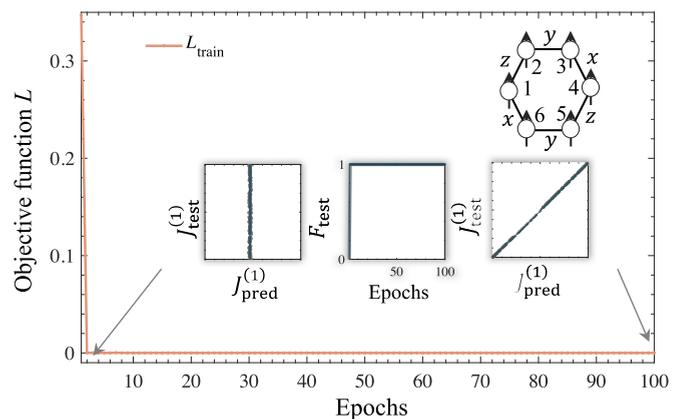


FIG. 7. Trained results for the 6-qubit Kitaev model on the honeycomb lattice $\mathcal{H}_{\text{Kit}}^6$. The predicted and actual values for the parameters $J^{(1)}$ of 100 test samples are compared.

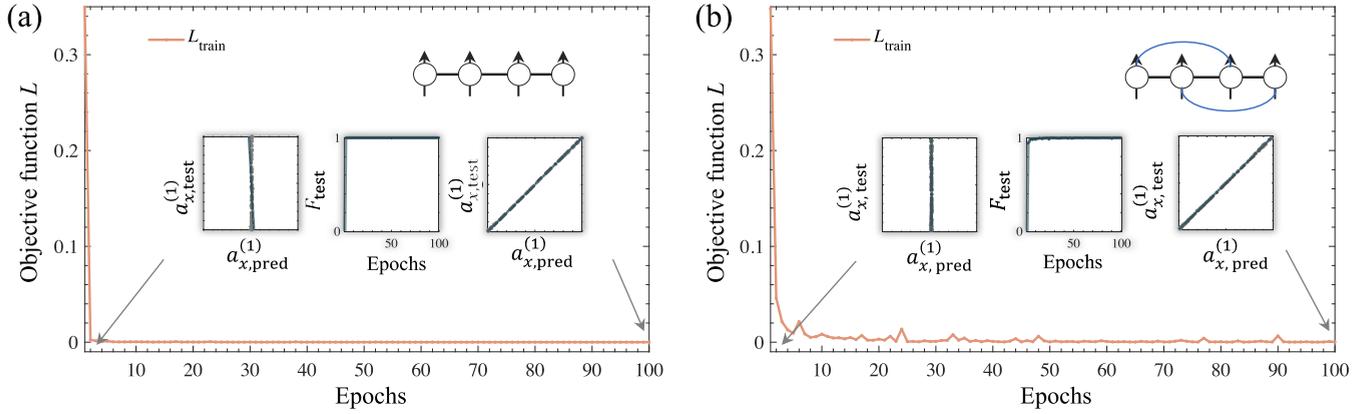


FIG. 8. Trained results for 4-qubit integrable Hamiltonians $\mathcal{H}_{\text{int}}^4$ (a) and the nonintegrable Hamiltonians $\mathcal{H}_{\text{nint}}^4$ (b). The objective function L_{train} and the similarity F_{test} are shown. The comparison between the predicted and actual values for the parameters $a_x^{(1)}$ is also presented by testing 100 samples.

of single-qubit measurements even without local terms due to the coupling between qubits.

Last, we discuss the differences when applying our method to the integrable and nonintegrable Hamiltonians [45]. In the simulation, the considered integrable and nonintegrable Hamiltonians are written as

$$\mathcal{H}_{\text{int}}^4 = \sum_{i=1}^4 a_x^{(i)} \sigma_x^{(i)} + \sum_{j=1}^3 J^{(j)} \sigma_z^{(j)} \sigma_z^{(j+1)}, \quad (8)$$

$$\mathcal{H}_{\text{nint}}^4 = \sum_{i=1}^4 a_x^{(i)} \sigma_x^{(i)} + \sum_{j=1}^3 J_1^{(j)} \sigma_z^{(j)} \sigma_z^{(j+1)} + \sum_{j=1}^2 J_2^{(j)} \sigma_z^{(j)} \sigma_z^{(j+2)}. \quad (9)$$

The trained results are shown in Fig. 8, including the loss functions L_{train} and learning accuracy F_{test} as a function of epochs. From these results, there are no big differences in the aspect of the convergence speed and the learning accuracy, when applying our method to the integrable and nonintegrable Hamiltonians.

VI. CONCLUSIONS

We conclude that a composite neural network can be trained to learn the Hamiltonians from single-qubit measurements, and numerical simulations of up to 7 qubits have demonstrated its feasibility on time-independent and time-dependent Hamiltonians. Compared with the existing methods, this neural network method does not need to prepare the eigenstates of target Hamiltonians and it can learn all the information of Hamiltonians including the magnitude and sign of the parameters. Once the neural network is successfully trained, it can be directly used to learn the unknown Hamiltonians from the measured data without any post-processing. It is a “once for all” advantage. Besides, single-qubit measurements are easy-to-implemented in current quantum platforms [13,46–49], such as using the dispersive readout on superconducting qubits and the ensemble measurements on nuclear magnetic resonance. Single-qubit measurements also have the lower readout errors than multi-qubit measurements [50,51]. The high accuracy can be achieved even under the potential experimental noises, includ-

ing Gaussian noise and decoherence effect. It will bring some potential applications in performing the tasks of Hamiltonians identification in the experiments. Our method also has possible extensions in the future, such as learning the environment information around the system and simulating the dynamics of closed and open systems.

ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China (Grant No. 2019YFA0308100), National Natural Science Foundation of China (Grants No. 12075110, No. 11975117, No. 11905099, No. 11875159, and No. U1801661), Guangdong Basic and Applied Basic Research Foundation (2019A1515011383), Guangdong International Collaboration Program (Grant No. 2020A0505100001), Guangdong Provincial Key Laboratory (2019B121203002), Science, Technology, and Innovation Commission of Shenzhen Municipality (Grants No. ZDSYS20170303165926217, No. KQTD20190929173815000, No. JCYJ20200109140803865, No. JCYJ20170412152620376, and No. JCYJ20180302174036418), and Pengcheng Scholars, Guangdong Innovative and Entrepreneurial Research Team Program (Grant No. 2019ZT08C044).

APPENDIX A: DYNAMICS OF SINGLE-QUBIT MEASUREMENTS

We consider that a quantum system with N qubits starts from an initial state $|\Psi_0\rangle$, undergoes a dynamical process governed by the unknown Hamiltonian \mathcal{H} . \mathcal{H} is parameterized as

$$\mathcal{H} = \sum_{m=1}^M a_m P_m, \quad (A1)$$

where P_m is the tensor product of Pauli matrices I , σ_x , σ_y , and σ_z , and a_m is the parameter of Hamiltonians. If we measure single-qubit Pauli observable $P \in \{\sigma_{x,y,z}^{(i)}\}$, its expectation value is

$$\bar{P}(t) = \langle \Psi_0 | P(t) | \Psi_0 \rangle, P(t) = e^{i\mathcal{H}t} P e^{-i\mathcal{H}t}. \quad (A2)$$

According to

$$e^{i\lambda G} A e^{-i\lambda G} = A + i\lambda [G, A] + \frac{i^2 \lambda^2}{2!} [G, [G, A]] + \dots + \frac{i^n \lambda^n}{n!} [G, [G, \dots [G, A] \dots]] + \dots, \quad (\text{A3})$$

we get

$$\bar{P}(t) = \bar{P} + \sum_{n=1}^{\infty} \frac{i^n t^n}{n!} \sum_{i_1=1}^M \sum_{i_2=1}^M \dots \sum_{i_n=1}^M f_{i_1 i_2 \dots i_n} a_{i_1} a_{i_2} \dots a_{i_n}. \quad (\text{A4})$$

Here, a_{i_n} is the parameter of Hamiltonian in Eq. (A1) and the coefficient $f_{i_1 i_2 \dots i_n}$ is

$$f_{i_1 i_2 \dots i_n} = \langle \Psi_0 | [B_{i_1}, [B_{i_2}, \dots [B_{i_n}, P]]] | \Psi_0 \rangle. \quad (\text{A5})$$

It means that, if $f_{i_1 i_2 \dots i_n}$ is not zero, the parameter $a_{i_1} a_{i_2} \dots a_{i_n}$ participates in the dynamics of single-qubit observables and it is possible to learn the Hamiltonian from the temporal records of the expectation values of P . For this purpose, two key points are carefully made in our method. First, we choose $|\Psi_0\rangle = \prod_{i=1}^N \otimes |\psi_0^i\rangle$ as the initial state, with $|\psi_0^i\rangle = R_z(\pi/4)R_y(\pi/4)|0\rangle$. This state has nonzero projections on x , y , and z directions, which will ensure $f_{i_1 i_2 \dots i_n}$ is not zero as much as possible, and this state is easy-to-prepared in quantum platforms. Second, we choose to measure the dynamics of single-qubit Pauli observables, which is also easy-to-implemented in the experiments compared with the multiqubit measurements. Under such an initial state and single-qubit measurements, not all the Hamiltonians can be learned, but the most common of Hamiltonians can be estimated from the dynamics of $\bar{P}(t)$.

APPENDIX B: DETAILED FRAMEWORK OF RNN

As shown in Fig. 1 in the main text, in our NN framework, we use long short-term memory (LSTM) network, which is a type of RNN [52]. Compared with traditional feed-forward neural networks, LSTM can learn the correlation in time sequences, which has been widely applied on handwriting recognition and speech recognition in the classical field [53], and quantum control and quantum process tomography in the quantum field [54,55]. So, LSTM is appropriate to estimate the Hamiltonians from the temporal records of single-qubit measurements. In this training, we define the input and output layers, objective function, and similarity function as follows:

(i) The input and output layers: At the moment $s\tau$, the expectation values of single-qubit measurements are collected as a vector

$$\mathbf{O}(s\tau) = \{O_k^{(i)}(s\tau) | k = x, y, z, 1 \leq i \leq N\}. \quad (\text{B1})$$

It is firstly fed into the s th LSTM cell. $\mathbf{I} = \{\mathbf{O}(s\tau) | 1 \leq s \leq S\}$ and $\mathbf{H} = \{a_m | 1 \leq m \leq M\}$ are respectively used as the input and output layers of our NN framework. Hence, the number of required LSTM cells equals the number of sampling points S . The structures for exporting \mathbf{H} are different for time-dependent and time-independent Hamiltonians. For time-dependent parameters in Hamiltonians, f_s is imported to a composite neural network including LSTM and FC neural networks (path A). Repetitive LSTM cells decode the vector f_s and FC neural networks project the output of each cell

to a series of time-dependent Hamiltonian parameters. For time-independent parameters in Hamiltonians, an FC neural network directly follows the LSTM cells. Here, the FC neural networks do not have hidden layers.

(ii) The objective function: Our neural network is trained by minimizing the distance between the predicted outcome \mathbf{H}^{pred} and the true outcome \mathbf{H}^{true} . Here, we use the mean square error (MSE) between \mathbf{H}^{pred} and \mathbf{H}^{true} as the objective function. It is

$$L = \frac{1}{M} \sum_{m=1}^M (\mathbf{H}_m^{\text{true}} - \mathbf{H}_m^{\text{pred}})^2. \quad (\text{B2})$$

This definition can learn the magnitude and sign of the parameters, because L decreases to 0 only when $\mathbf{H}_m^{\text{true}}$ and $\mathbf{H}_m^{\text{pred}}$ are absolutely the same. In order to minimize the objective function in this work, we use Adam optimization algorithm, one of the state-of-the-art gradient descent algorithms, to train the hidden parameters of the network.

(iii) The similarity function: In order to estimate the performance of our trained NN, we need to compute the similarity between the predicted and the real outcomes for the test data. Here, we use the definition of the cosine proximity function between two vectors. It is

$$F(\mathbf{H}^{\text{pred}}, \mathbf{H}) = \frac{(\mathbf{H}^{\text{pred}} \cdot \mathbf{H})}{(\|\mathbf{H}^{\text{pred}}\| \cdot \|\mathbf{H}\|)}. \quad (\text{B3})$$

APPENDIX C: STRUCTURE OF LSTM CELL

The LSTM is a form of the recurrent neural network designed to solve the long-term dependencies problem. An LSTM consists of a chain of repeating neural network modules called LSTM cells. As shown in Fig. 9(a), the s th LSTM cell imports $O(s\tau)$, f_{s-1} , and c_{s-1} and exports f_s and c_s for the next LSTM cell. Here, $O(s\tau)$ and f_{s-1} are firstly combined by an FC neural network whose structure is shown in Fig. 9(b). In our training, this layer includes 256 neurons. Then different activation functions σ and \tanh are used and finally different operations \oplus and \otimes are implemented before exporting f_s and c_s . Next, we introduce the detailed operations in the LSTM cell.

As shown in Fig. 9(a), the long-term memory of LSTM is called cell state c_s , which stores information learned by flowing through the entire chain. To update the cell state, the cell has two layers called “forget gate” and “input gate” to remove or add information to the cell state. The cell also has the ability to output the information from cell state called “output gate”. Thus, these three gates control the cell state and construct an LSTM cell. At the beginning, the cell uses forget gate G to decide what past information to remove from the cell. The input of current moment $O(s)$ and the output of last moment f_{s-1} go through the forget gate G as follows:

$$G = \sigma(W_g \cdot [f_{s-1}, O(s)]^T + b_g), \quad (\text{C1})$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the Sigmoid function. Then, it uses input gate I to decide what new information to add to the cell state as follows: $I = \sigma(W_i \cdot [f_{s-1}, O(s)]^T + b_i)$. And $O(s)$ and f_{s-1} go through a tanh layer to create a candidate cell state

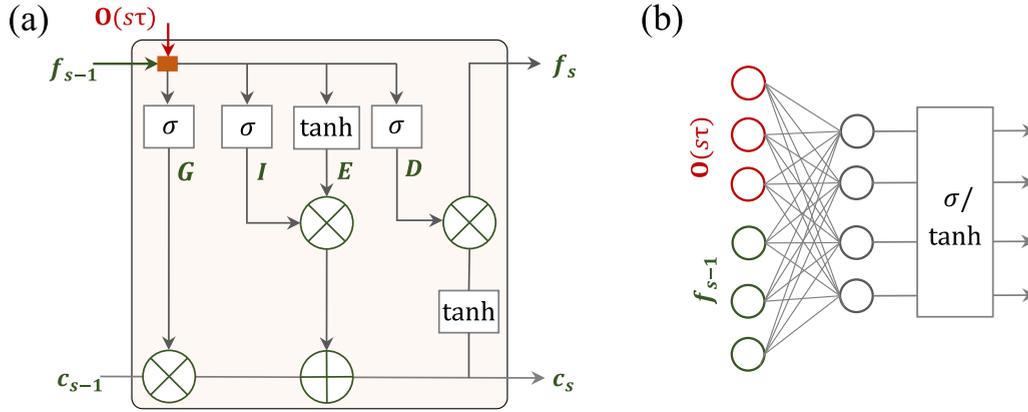


FIG. 9. The schematic diagram for LSTM [(a), (b)]. The right plot presents the operation combining the input f_{s-1} and $O(s\tau)$ (labeled by red square) with one layer including 256 neurons.

E as follows:

$$E = \tanh(W_e \cdot [f_{s-1}, O(s)]^T + b_e). \quad (\text{C2})$$

The next step is to update the cell state by forget gate G and input gate I as follows: $c_s = G \times f_{s-1} + I \times E$. In the end, it uses output gate to decide what information to select as output and generate the output. The equations are given as: $D = \sigma(W_d \cdot [f_{s-1}, O(s)]^T + b_d)$ and $f_s = D \times \tanh(c_s)$.

APPENDIX D: DETAILS ABOUT ROBUSTNESS SIMULATIONS

Robustness simulations are performed for a 3-qubit system with Ising Hamiltonian \mathcal{H}_{XY}^3 . \mathcal{H}_{XY}^3 is

$$\mathcal{H}_{XY}^3 = \sum_{i=1}^3 a_z^{(i)} \sigma_z^{(i)} + \sum_{j=1}^2 J^{(j)} (\sigma_x^{(j)} \sigma_x^{(j+1)} + \sigma_y^{(j)} \sigma_y^{(j+1)}). \quad (\text{D1})$$

The unknown parameters in \mathcal{H}_{XY}^3 form a vector $\mathbf{H} = [a_z^{(1)}, a_z^{(2)}, a_z^{(3)}, J^{(1)}, J^{(2)}]^T$ as the output of RNN. The expectation values of single-qubit observables $\sigma_{x,y,z}^{(i)}$ are measured at a discrete time separated by $\tau = 0.02\pi/J_0$, and they are collected as the input data \mathbf{I} of RNN.

Robustness simulations under Gaussian noise—In the main text, we respectively train the neural networks using the noiseless and Gaussian-noise (noise level $\epsilon = 0.1$) data and then use trained models (RNN_0noise_25, RNN_10noise_50, RNN_0noise_25, and RNN_10noise_50) to learn Hamiltonians from the noisy test data. These noisy data is artificially generated by adding the Gaussian noise in the measurement data \mathbf{I} , i.e., $\mathbf{I}' = \mathbf{I} + \mathcal{N}(0, \epsilon)$. Here, $\mathcal{N}(0, \epsilon)$ is a Gaussian distribution with the mean of 0 and the standard deviation of ϵ . We change ϵ from 2% to 10% with the step 2% and

create 5000 noisy test data for each ϵ to test the performance of different models. From the simulations, it can be roughly concluded that learning Hamiltonians via RNN has robust performance under the Gaussian noise.

Robustness simulations under decoherence effect—In the main text, we also train the neural networks using the noiseless and decoherence data (models RNN_0noise_150 and RNN_t2noise_150). The training and test data with decoherence effect are created according to the Kraus representation of decoherence dynamics. The evolution of Hamiltonians is divided into slices with the duration of each slice being $\delta\tau$. Supposing that the density matrix is $\rho(t)$ at the moment t , then density matrix at $t + \delta\tau$ is

$$\rho(t + \delta\tau) = \sum_{i=1}^3 \sum_{j=0}^1 E_j^i e^{-i\mathcal{H}\delta\tau} \rho(t) e^{i\mathcal{H}\delta\tau} E_j^{i\dagger}. \quad (\text{D2})$$

Here, E_j^i is the kraus operator of the i th qubit with,

$$E_0^i = \sqrt{\lambda_i} I_2, E_1^i = \sqrt{1 - \lambda_i} \sigma_z^i \quad (\text{D3})$$

λ_i is a parameter with $\lambda_i = (1 + e^{-\delta\tau/T_2^i})/2$. T_2^i is the decoherence time of i -th qubit. We change T_2^i from $1\pi/J_0$ to $6\pi/J_0$ with the segment $2\pi/J_0$. For each T_2^i , we create 5000 decoherence test data with the sample points of $S = 150$ (sample interval is $0.02\pi/J_0$ and corresponding sampling time is $3\pi/J_0$). As shown in Fig. 4(b) in the main text, when we feed these test data to the model RNN_0noise_150 to predict the Hamiltonian parameters \mathbf{H} , it is found that the accuracy of predicted \mathbf{H} rapidly falls with the decrease of coherence time. However, the robustness against the decoherence effect is significantly improved when using the model RNN_T2noise_150.

- [1] V. Schäfer, C. Ballance, K. Thirumalai, L. Stephenson, T. Ballance, A. Steane, and D. Lucas, Fast quantum logic gates with trapped-ion qubits, *Nature* **555**, 75 (2018).
 [2] M. Mohseni, A. T. Rezakhani, and D. A. Lidar, Quantum-process tomography: Resource analysis of different strategies, *Phys. Rev. A* **77**, 032322 (2008).

- [3] J. Liu, Y. Qi, Z. Y. Meng, and L. Fu, Self-learning monte carlo method, *Phys. Rev. B* **95**, 041101(R) (2017).
 [4] L. Huang and L. Wang, Accelerated monte carlo simulations with restricted Boltzmann machines, *Phys. Rev. B* **95**, 035105 (2017).

- [5] J. Liu, H. Shen, Y. Qi, Z. Y. Meng, and L. Fu, Self-learning Monte Carlo method and cumulative update in fermion systems, *Phys. Rev. B* **95**, 241104(R) (2017).
- [6] Y. Nagai, H. Shen, Y. Qi, J. Liu, and L. Fu, Self-learning Monte Carlo method: Continuous-time algorithm, *Phys. Rev. B* **96**, 161102(R) (2017).
- [7] H. Shen, J. Liu, and L. Fu, Self-learning monte carlo with deep neural networks, *Phys. Rev. B* **97**, 205140 (2018).
- [8] J. B. Rigo and A. K. Mitchell, Machine learning effective models for quantum systems, *Phys. Rev. B* **101**, 241105(R) (2020).
- [9] N. D. Mermin, *Quantum Computer Science: An Introduction* (Cambridge University Press, Cambridge, 2007).
- [10] D. Dong and I. R. Petersen, Quantum control theory and applications: A survey, *IET Control Theory & Applications* **4**, 2651 (2010).
- [11] J. Helsen, I. Roth, E. Onorati, A. H. Werner, and J. Eisert, A general framework for randomized benchmarking, [arXiv:2010.07974](https://arxiv.org/abs/2010.07974).
- [12] I. M. Georgescu, S. Ashhab, and F. Nori, Quantum simulation, *Rev. Mod. Phys.* **86**, 153 (2014).
- [13] T. Xin, Improved quantum state tomography for systems with $xx+yy$ couplings and z readouts, *Phys. Rev. A* **102**, 052410 (2020).
- [14] T. Xin, D. Lu, J. Klassen, N. Yu, Z. Ji, J. Chen, X. Ma, G. Long, B. Zeng, and R. Laflamme, Quantum State Tomography Via Reduced Density Matrices, *Phys. Rev. Lett.* **118**, 020401 (2017).
- [15] C. Di Franco, M. Paternostro, and M. S. Kim, Hamiltonian Tomography In An Access-Limited Setting Without State Initialization, *Phys. Rev. Lett.* **102**, 187203 (2009).
- [16] J. H. Cole, S. G. Schirmer, A. D. Greentree, C. J. Wellard, D. K. L. Oi, and L. C. L. Hollenberg, Identifying an experimental two-state hamiltonian to arbitrary accuracy, *Phys. Rev. A* **71**, 062312 (2005).
- [17] S. J. Devitt, J. H. Cole, and L. C. L. Hollenberg, Scheme for direct measurement of a general two-qubit hamiltonian, *Phys. Rev. A* **73**, 052317 (2006).
- [18] J. Zhang and M. Sarovar, Quantum Hamiltonian Identification From Measurement Time Traces, *Phys. Rev. Lett.* **113**, 080401 (2014).
- [19] J. Zhang and M. Sarovar, Identification of open quantum systems from observable time traces, *Phys. Rev. A* **91**, 052121 (2015).
- [20] S.-Y. Hou, H. Li, and G.-L. Long, Experimental quantum hamiltonian identification from measurement time traces, *Sci. Bull.* **62**, 863 (2017).
- [21] A. Sone and P. Cappellaro, Hamiltonian identifiability assisted by a single-probe measurement, *Phys. Rev. A* **95**, 022335 (2017).
- [22] A. Sone and P. Cappellaro, Exact dimension estimation of interacting qubit systems assisted by a single quantum probe, *Phys. Rev. A* **96**, 062334 (2017).
- [23] X.-L. Qi and D. Ranard, Determining a local hamiltonian from a single eigenstate, *Quantum* **3**, 159 (2019).
- [24] M. Dupont, N. Macé, and N. Laflorencie, From eigenstate to hamiltonian: Prospects for ergodicity and localization, *Phys. Rev. B* **100**, 134201 (2019).
- [25] J. R. Garrison and T. Grover, Does A Single Eigenstate Encode The Full Hamiltonian?, *Phys. Rev. X* **8**, 021026 (2018).
- [26] E. Bairey, I. Arad, and N. H. Lindner, Learning A Local Hamiltonian From Local Measurements, *Phys. Rev. Lett.* **122**, 020504 (2019).
- [27] T. Xin, S. Lu, N. Cao, G. Anikeeva, D. Lu, J. Li, G. Long, and B. Zeng, Local-measurement-based quantum state tomography via neural networks, *Npj Quantum Inf.* **5**, 109 (2019).
- [28] Z. Li, L. Zou, and T. H. Hsieh, Hamiltonian Tomography Via Quantum Quench, *Phys. Rev. Lett.* **124**, 160502 (2020).
- [29] B. S. Rem, N. Käming, M. Tarnowski, L. Asteria, N. Fläschner, C. Becker, K. Sengstock, and C. Weitenberg, Identifying quantum phase transitions using artificial neural networks on experimental data, *Nat. Phys.* **15**, 917 (2019).
- [30] E. P. Van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Learning phase transitions by confusion, *Nat. Phys.* **13**, 435 (2017).
- [31] P. Huembeli, A. Dauphin, and P. Wittek, Identifying quantum phase transitions with adversarial neural networks, *Phys. Rev. B* **97**, 134109 (2018).
- [32] J. F. Rodriguez-Nieva and M. S. Scheurer, Identifying topological order through unsupervised machine learning, *Nat. Phys.* **15**, 790 (2019).
- [33] W. Lian, S.-T. Wang, S. Lu, Y. Huang, F. Wang, X. Yuan, W. Zhang, X. Ouyang, X. Wang, X. Huang *et al.*, Machine Learning Topological Phases With A Solid-State Quantum Simulator, *Phys. Rev. Lett.* **122**, 210503 (2019).
- [34] S. Lu, S. Huang, K. Li, J. Li, J. Chen, D. Lu, Z. Ji, Y. Shen, D. Zhou, and B. Zeng, Separability-entanglement classifier via machine learning, *Phys. Rev. A* **98**, 012315 (2018).
- [35] C. Harney, S. Pirandola, A. Ferraro, and M. Paternostro, Entanglement classification via neural network quantum states, *New J. Phys.* **22**, 045001 (2020).
- [36] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Neural-network quantum state tomography, *Nat. Phys.* **14**, 447 (2018).
- [37] S. Ahmed, C. S. Muñoz, F. Nori, and A. F. Kockum, Quantum state tomography with conditional generative adversarial networks, [arXiv:2008.03240](https://arxiv.org/abs/2008.03240).
- [38] E. Magesan, J. M. Gambetta, A. D. Córcoles, and J. M. Chow, Machine Learning For Discriminating Quantum Measurement Trajectories And Improving Readout, *Phys. Rev. Lett.* **114**, 200501 (2015).
- [39] M. Khanahmadi and K. Mølmer, Time-dependent atomic magnetometry with a recurrent neural network, *Phys. Rev. A* **103**, 032406 (2021).
- [40] V. Cimini, M. Barbieri, N. Treps, M. Walschaers, and V. Parigi, Neural Networks for Detecting Multimode Wigner-Negativity, *Phys. Rev. Lett.* **125**, 160504 (2020).
- [41] X. Ma, Z. C. Tu, and S.-J. Ran, Deep neural network predicts parameters of quantum many-body hamiltonians by learning visualized wave-functions, [arXiv:2012.03019](https://arxiv.org/abs/2012.03019).
- [42] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)* (ICLR, 2015).
- [43] T. Minakawa, J. Nasu, and A. Koga, Quantum and classical behavior of spin- s Kitaev models in the anisotropic limit, *Phys. Rev. B* **99**, 104408 (2019).
- [44] H. Takagi, T. Takayama, G. Jackeli, G. Khaliullin, and S. E. Nagler, Concept and realization of Kitaev quantum spin liquids, *Nat. Rev. Phys.* **1**, 264 (2019).
- [45] X. Nie, B.-B. Wei, X. Chen, Z. Zhang, X. Zhao, C. Qiu, Y. Tian, Y. Ji, T. Xin, D. Lu, and J. Li, Experimental Observation

- Of Equilibrium And Dynamical Quantum Phase Transitions Via Out-Of-Time-Ordered Correlators, *Phys. Rev. Lett.* **124**, 250601 (2020).
- [46] D. Keith, M.G. House, M.B. Donnelly, T.F. Watson, B. Weber, and M.Y. Simmons, Single-Shot Spin Readout In Semiconductors Near The Shot-Noise Sensitivity Limit, *Phys. Rev. X* **9**, 041003 (2019).
- [47] C. D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage, Trapped-ion quantum computing: Progress and challenges, *Appl. Phys. Rev.* **6**, 021314 (2019).
- [48] D. M. Irber, F. Poggiali, F. Kong, M. Kieschnick, T. Lühmann, D. Kwiatkowski, J. Meijer, J. Du, F. Shi, and F. Reinhard, Robust all-optical single-shot readout of nitrogen-vacancy centers in diamond,, *Nat. Commn.* **12**, 532 (2021).
- [49] T. Xin, B.-X. Wang, K.-R. Li, X.-Y. Kong, S.-J. Wei, T. Wang, D. Ruan, and G.-L. Long, Nuclear magnetic resonance for quantum computing: Techniques and recent achievements, *Chin. Phys. B* **27**, 020308 (2018).
- [50] J. Gambetta, W. A. Braff, A. Wallraff, S. M. Girvin, and R. J. Schoelkopf, Protocols for optimal readout of qubits using a continuous quantum nondemolition measurement, *Phys. Rev. A* **76**, 012325 (2007).
- [51] B. Nachman, M. Urbanek, W. A. de Jong, and C. W. Bauer, Unfolding quantum computer readout noise, *npj Quantum Inf.* **6**, 84 (2020).
- [52] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Comput.* **9**, 1735 (1997).
- [53] H. Sak, A. Senior, and F. Beaufays, Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition, [arXiv:1402.1128](https://arxiv.org/abs/1402.1128).
- [54] L. Banchi, E. Grant, A. Rocchetto, and S. Severini, Modelling non-markovian quantum processes with recurrent neural networks, *New J. Phys.* **20**, 123030 (2018).
- [55] E. Flurin, L. S. Martin, S. Hacothen-Gourgy, and I. Siddiqi, Using A Recurrent Neural Network To Reconstruct Quantum Dynamics Of A Superconducting Qubit From Physical Observations, *Phys. Rev. X* **10**, 011006 (2020).